2. What does insecure direct object reference mean?

Insecure Direct Object Reference (IDOR) means that a web application allows users to access or manipulate resources directly by referencing an internal object, such as a file, record, or user ID, without sufficient security checks to confirm that the user has permission to do so. In other words, **the application exposes these internal references to users without adequately verifying access permissions**, making it possible for an unauthorized user to access or alter data simply by modifying these references.

Breaking It Down

- **"Insecure"**: There's a lack of adequate security controls.
- **"Direct Object Reference"**: The application directly uses internal identifiers (like IDs) to reference objects, such as database records, files, or user accounts, in URLs or parameters.

How IDOR Works

Imagine a situation where a user profile page URL looks like:

```
https://example.com/user?profile_id=123
```

If the application doesn't verify that `profile_id=123` actually belongs to the logged-in user, then a malicious user could change this to `profile_id=456` to view, modify, or delete another user's profile.

Why IDOR is a Problem

An IDOR vulnerability makes it easy for unauthorized users to gain access to or manipulate sensitive information. This flaw arises because **developers allow access to objects based solely on input parameters (like IDs)** without verifying permissions or ownership on the server.

Examples of IDOR Scenarios

1. **Account Access**: Changing `account_id` in a banking app URL to view another user's account information.
2. **File Downloads**: Accessing someone else's files by changing `file_id` in a document management system.
3. **Order Details**: Modifying an `order_id` to view or edit another user's order history.

Mitigating IDOR Vulnerabilities

To prevent IDOR, applications should:

- **Use Server-Side Authorization Checks**: Always confirm that a user has permission to access or modify an object.
- **Limit Exposure of Object References**: Use unpredictable references (like UUIDs) instead of sequential or guessable IDs.
- **Conduct Security Audits**: Regularly test for access control vulnerabilities as part of the development and deployment process.

IDOR is an essential concept in web security and falls under **Broken Access Control** vulnerabilities, often leading to serious privacy and data integrity issues.